Group Assignment

CS 441/541: Artificial Intelligence

Project Title :

Comparative Analysis of Various Language Models for Question Answering on the SQuAD Dataset

Team Members:

Pranav Dharamthok - pran4@pdx.edu

Upasana Chaudhari - upasana@pdx.edu

Chethana Muppalam - chetha@pdx.edu

Varshini Puttaswamy Ballari - varshi@pdx.edu

Code Repository:

https://github.com/chethana613/qna-ai-chatbot

Table of Contents:

Team Contribution

This project was a collaborative effort, with each team member making significant contributions to ensure its success. The team's dedication and expertise were instrumental in achieving the project's objectives. The success of our project was a result of both individual expertise and collective collaboration. Chethana Muppalam took the lead in fine-tuning the RoBERTa model, ensuring optimal performance, and meticulously documenting the process. Pranav Dharamthok focused on the T5 model, achieving robust performance enhancements and providing comprehensive documentation. Upasana Chaudhari dedicated her efforts to fine-tuning the BERT model, resulting in significant improvements, and thoroughly documented her findings. Varshini Puttaswamy Ballari worked on the DistilBERT model, enhancing its efficiency, and carefully documented the procedures and results.

While each member brought their unique skills to fine-tune and document a specific model, the project was a true collaborative effort, with team members supporting each other and integrating their work seamlessly to achieve our collective goal.

1. Introduction

Question answering is the discipline which aims to build systems that automatically answer questions posed by humans in a natural language. In the extractive question answering paradigm, the answers to a question are spans of text extracted from a single document. In the SQuAD benchmark Dataset, each answer lies in a paragraph from Wikipedia.

In the open-domain setting, the answers are sought in a large collection of texts such as the whole English Wikipedia. State-of-the-art performances in usual Question Answering are achieved thanks to powerful and heavy pretrained language models that rely on sophisticated attention mechanisms and hundreds of millions of parameters. Attention mechanisms are key components of such systems since they allow building contextualized and questionaware representations of the words in the documents and extract the span of text which is most likely the correct answer. These models are very resource-demanding and need GPUs to be scalable.

Our project focuses on evaluating the performance of various state-of-the-art natural language processing (NLP) models on the SQuAD dataset. Specifically, we have fine-tuned and assessed the models BERT, RoBERTa, DistilBERT, and T5 to determine their effectiveness in answering questions based on Wikipedia articles.

SQuAD, short for Stanford Question Answering Dataset, is a well-established benchmark in the field of natural language processing (NLP). It contains over 100,000 question-answer pairs derived from more than 500 Wikipedia articles. Each question in the dataset is accompanied by a corresponding answer, which is a specific span of text

within the related article. This structure makes SQuAD an ideal dataset for training and evaluating question answering systems.

In our study, we fine-tuned BERT, RoBERTa, DistilBERT, and T5 models on the SQuAD dataset. We then assessed their performance by comparing their answers to the reference answers provided in the dataset. Our goal was to measure the effectiveness of these models in terms of accuracy, F1 score, ROUGE score, and BLEU score.

By conducting this comparative analysis, we aim to provide insights into the strengths and weaknesses of each model in the context of question answering. This study not only highlights the capabilities of these transformer models but also contributes to the ongoing research in natural language understanding and the development of more advanced question answering systems.

2. Data Description

SQuAD is composed of questions posed by crowdworkers on a set of Wikipedia articles. The answers to these questions are segments of text, or spans, from the corresponding reading passages. In some cases, the questions might be unanswerable, requiring systems to determine when no answer is supported by the provided paragraph and to abstain from answering.

We have used SQuAD 1.1 - SQuAD 1.1 contains 107,785 question-answer pairs on 536 articles. In SQuAD, the correct answers of questions can be any sequence of tokens in the given text.

Dataset Link : https://huggingface.co/datasets/rajpurkar/squad

Dataset Split: The SQuAD dataset is divided into training and validation sets. The training set consists of 87,599 samples, while the validation set contains 20,302 samples. Given the computational resources available for our project, we have utilized a subset of 10,000 samples for training and 1,000 samples for validation. This subset allows us to efficiently fine-tune and evaluate various NLP models while maintaining a manageable computational load.

Dataset Attributes:

- **id** : A unique identifier for each question-answer pair in the dataset. This field helps in tracking and referencing specific entries within the dataset.
- **title:** The title of the Wikipedia article from which the context is extracted. It provides context for the passage and helps in identifying the source of the information.
- **context:** A passage from a Wikipedia article that contains the information needed to answer the corresponding question. The context is typically a paragraph or a few paragraphs long and serves as the source text for deriving answers.
- **question:** A question posed by a crowdworker based on the given context. The question can either be answerable with a specific span of text from the context or unanswerable, requiring the system to recognize the lack of sufficient information in the context.
- **answers:** this field contains the answer(s) to the question. This field has the following subfields:

- **text**: A list of strings representing the answer text(s) extracted from the context. In cases where there are multiple correct answers, this list will contain all valid answers.
- **answer_start**: A list of integers indicating the starting character positions of each answer span within the context. Each position corresponds to the beginning of an answer in the context text.

3. Models Used

3.1. BERT

BERT (Bidirectional Encoder Representations from Transformers) is a groundbreaking technique in the field of natural language processing (NLP) developed by Google. It is designed to understand the context of a word in search queries, making it extremely effective for tasks like question answering and language inference. BERT's bidirectional approach means it reads text from both left to right and right to left, enabling it to understand the full context of a word by looking at the words before and after it.

Fine-tuning is the process of taking a pre-trained model and adjusting it to perform specific tasks.

3.2. RoBERTa

RoBERTa (Robustly Optimized BERT Approach) is a state-of-the-art language representation model developed by Facebook AI. It is based on the original BERT (Bidirectional Encoder Representations from Transformers) architecture but differs in several key ways. RoBERTa’s objective is to improve the original BERT model by

expanding the model, the training corpus, and the training methodology to better utilize the Transformer architecture. This produces a representation of language that is more expressive and robust, which has been shown to achieve state-of-the-art performance on a wide range of NLP tasks. This model is trained on a large amount of text data from multiple languages, which makes it capable of understanding and generating text in different languages.

3.3. DistilBERT

DistilBERT, short for Distilled Bidirectional Encoder Representations from Transformers, is a compact version of BERT developed by Hugging Face, aiming to reduce computational resources while preserving performance. It employs knowledge distillation from a larger BERT model, retaining BERT's bidirectional approach to contextual understanding of words. This capability allows DistilBERT to grasp the complete context of a word by considering both preceding and following words in a sentence or document. Like BERT, DistilBERT is fine-tuned for specific NLP tasks such as text classification and named entity recognition, adapting its parameters to achieve optimal performance on these tasks. DistilBERT thus provides a more resource-efficient solution for various NLP applications, maintaining robustness in natural language understanding.

3.4. T5

T5 (Text-to-Text Transfer Transformer) models are designed to convert all NLP problems into a text-to-text format, where both the input and output are always text

strings. The T5 models are pre-trained on a multi-task mixture of unsupervised and supervised tasks, using a span-corruption objective designed to mirror the downstream tasks as closely as possible.

3.4.1. T5 Small

This variant is the smallest, featuring fewer parameters (around 60 million) compared to other versions. It is faster to run and requires less computational resources but generally performs with lower accuracy and overall performance metrics than its larger counterparts.

3.4.2. T5 Base

With approximately 220 million parameters, the T5 Base model strikes a balance between computational efficiency and performance. It is suitable for most tasks and offers significantly better performance than the Small model while still being reasonably efficient in terms of computational resources.

3.4.3. T5 Large

This variant is among the more robust versions of the T5, having about 770 million parameters. It offers high performance across a range of NLP tasks due to its greater capacity to capture complex patterns and relationships in the data. However, it requires more computational power and time to train or run.

4. Models Evaluation

Model evaluation involves assessing the performance of the trained model on a separate validation dataset. This helps in understanding how well the model generalizes to unseen data.

- Accuracy: Measures the overall correctness of the model's predictions.
- F1 Score: Considers both the precision and recall of the predictions to compute the score. It essentially measures the overlap between the predicted answers and the ground truth answers.
- BLEU (Bilingual Evaluation Understudy): Originally used for evaluating machine translation quality, it measures how many words and phrases in the prediction appear in the reference material. Higher scores indicate more overlap and typically better performance.
- ROUGE (Recall-Oriented Understudy for Gisting Evaluation): This evaluates text summaries by computing overlap scores such as ROUGE-1 (overlap of unigrams) and ROUGE-L (longest common subsequence).
- Exact Match (EM): Measures the percentage of predictions that match any of the ground truth answers exactly.

4.1. BERT

- During the training process, the model was trained for three epochs.
    - Epoch 1:Training Loss: 2.27, Epoch 2:Training Loss: 1.14, Epoch 3:Training Loss: 0.79

- The decrease in training loss over the epochs indicates that the model is learning and improving as it trains.
- The accuracy of the model is 44.7%. This means that the model correctly answered about 45% of the questions.
- The average F1 score is 0.082. This low score indicates that while the model makes some correct predictions, it struggles with finding all the relevant answers and avoiding incorrect ones.
- The low ROUGE scores indicate that the model's answers often do not match the wording of the correct answers.
- A BLEU score of 0.0 highlights a significant difference between the model's answers and the correct answers.

Model	Exact match	Accuracy	F1 Score	BLEU Score	ROUGE-1 Score	ROUGE-L Score
BERT	0.447	44.7	0.082	0.0	0.089	0.089

4.2. RoBERTa

- The RoBERTa-base fine-tuned model achieved an accuracy of 69.2%, with an exact match of 53% instances, suggesting moderate performance in answering or generating text.

- The model obtained moderate ROUGE-1 (0.0723) and ROUGE-L (0.071) scores, indicating some overlap in unigram and longer sequence matches between its outputs and reference texts.
- The model is trained for 2 epochs with Learning rate = 0.01 and Weight Decay = 0.01. The training loss decreased from 6.2841390712738034 in Epoch 1 to 6.244938175296784 in Epoch 2. This gradual decrease indicates that the model is progressively improving its ability to minimize the error between predicted outputs and actual targets as it continues to learn from the training data.
- F1 Score (0.0629) Suggests the model p to accurately identify and generate correct outputs or answers.
- BLEU Score (0.0189) Reflects a significant divergence between RoBERTa's generated text and human-generated references.

Model	Accuracy	Exact Match	F1 Score	BLEU Score	ROUGE-1 Score	ROUGE-L Score
RoBERTa-base	0.692	53	0.629	0.0189	0.0723	0.071

4.3. DistilBERT

- DistilBERT yielded an accuracy of 0.41 means that out of all predictions made by DistilBERT-base, 41% were correct. It's a fundamental metric to gauge how well the model performs in general.

- Exact Match score of 0.43 indicates that 43% of the time, DistilBERT-base provides answers that are completely identical to the correct answers provided in the dataset.
- A low F1 score of 0.052 for DistilBERT-base suggests that it is struggling with achieving both high precision and recall simultaneously, indicating potential challenges in model optimization.
- A BLEU score of 0.013 for DistilBERT-base indicates that the model's generated text has very little similarity to the reference translations used for evaluation. This metric is crucial for tasks like language generation and translation, where the goal is to produce accurate and fluent text.
- DistilBERT-base achieved a ROUGE-1 score of 0.064 and a ROUGE-L score of 0.0625, indicating moderate agreement with reference texts in terms of both individual words and longer sequences.

Model	Accuracy	Exact Match	F1 Score	BLEU Score	ROUGE-1 Score	ROUGE-L Score
DistilBERT-base	0.41	0.43	0.052	0.013	0.064	0.0625

4.4. T5

Model	Accuracy	Exact Match (%)	F1 Score	BLEU Score	ROUGE-1 Score	ROUGE-L Score
T5 Small	64.00%	76.4	83.879	0.290	0.799	0.799

T5 Base	70.90%	86.6	91.473	0.335	0.867	0.867
T5 Large	72.80%	88.4	93.636	0.309	0.885	0.885

The T5 models, which include Small, Base, and Large variants, have been evaluated on the SQuAD dataset, revealing distinct performance levels across various metrics. The T5 Small model achieved an Exact Match of 76.4% and an F1 score of 83.88%, with a BLEU score of 0.29, indicating a moderate overlap with reference answers. Its ROUGE scores, both for ROGUE-1 and ROUGE-L, hovered around 0.798, suggesting good unigram overlap and sequence matching. The T5 Base model demonstrated higher capabilities, with an Exact Match of 86.6% and an F1 score of 91.47%. Its BLEU score improved to 0.334, reflecting better word and phrase overlap compared to the Small variant, and its ROUGE score increased to about 0.867, indicating even better alignment with reference answers. The T5 Large model outperformed the others, recording the best performance with an Exact Match of 88.4% and an F1 score of 93.64%. Though its BLEU score was slightly lower at 0.309, its ROUGE scores were the highest at approximately 0.885, showcasing the best alignment and sequence matching with reference texts. This gradient of improvement highlights the advantages of increased model size and capacity in handling complex NLP tasks more effectively.

5. Challenges

- Training BERT requires substantial computational power and memory. Ensuring access to GPUs was crucial.

- Properly tokenizing and preparing the dataset was time-consuming and required careful handling to avoid data leakage.
- Fine-tuning could lead to overfitting, where the model performs well on training data but poorly on unseen data. Techniques like dropout and cross-validation were used to mitigate this.

6. Comparison of the Models Implemented

- **T5 models (Small, Base, and Large) consistently outperform** the other models across most metrics. The T5 Large model is the best performer overall.
- RoBERTa-base shows moderate performance, better than BERT, but significantly behind the T5 models.
- BERT has the lowest scores across most metrics, indicating it struggles with the task.
- DistilBERT has an excellent Exact Match score, but lacks comprehensive data for other metrics.

Why T5 Models Consistently Performed Well:

- The T5 models are pre-trained on a vast array of text generation and comprehension tasks, allowing them to generalize well.
- Increased model size from Small to Large enhances the ability to capture complex patterns, resulting in better performance.

- The T5 architecture is designed to handle sequence-to-sequence tasks effectively, which is crucial for tasks like text generation and summarization.

Moderately Performing Model(RoBERTa):

- RoBERTa-base performs moderately well due to effective fine-tuning on specific tasks, leading to decent accuracy and exact match scores. Its moderate scores suggest it is a reliable model for general tasks but may not be the best for tasks requiring high precision and text overlap with reference answers.
- Similar to T5, RoBERTa is pre-trained on large datasets, which helps in general NLP tasks but may not be as specialized for text generation as T5.

BERT and DistilBERT:

- BERT's training focuses on masked language modeling and next sentence prediction, which are not directly aligned with sequence generation tasks.
- Unlike T5 and RoBERTa, BERT is not designed for sequence-to-sequence tasks, limiting its effectiveness in tasks requiring coherent and contextually accurate text generation.
- The Metrics of DistilBERT explains for itself that this model is struggling with predicting answers from the context paragraphs of the squad dataset as it is primarily trained for text classification.

7. Future Scope

- Using larger and more diverse datasets can further improve the model's robustness and accuracy.
- Implementing techniques to reduce the model size and increase inference speed, making it more suitable for deployment in real-time applications.
- Investigating newer architectures such as GPT-3.5 turbo, Gemma, Llama and other transformer-based models to compare performance and capabilities.
- Experimenting with optimizers like ADAM, RMSprop, SGD etc., towards moderately performing models and analyzing their performance.

8. References

1. https://medium.com/red-buffer/building-a-simple-chatbot-with-llm-719a37659d30
2. https://pypi.org/project/t5/
3. https://huggingface.co/deepset/roberta-base-squad2
4. https://amitnikhade.medium.com/question-answering-in-association-with-roberta-a11518e70507
5. https://sh-tsang.medium.com/review-distilbert-a-distilled-version-of-bert-smaller-faster-cheaper-and-lighter-5b3fa180169e
6. https://towardsdatascience.com/distilbert-11c8810d29fc